Jonathan Skaggs

CS 478 Backprop

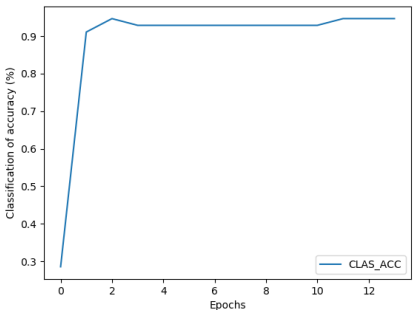
Iris Classification Problem

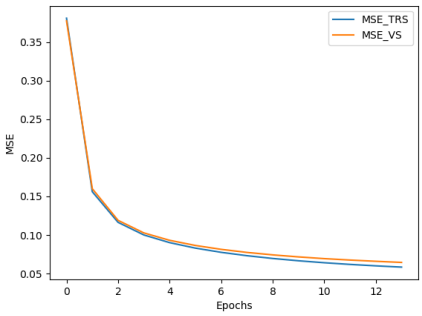
Stopping Criteria

I stopped when the Mean Squared Error (MSE) of the Validation Set (VS) stopped improving over a period of epochs (10). I keep track of the Best Solution So Far (BSSF) and update it after there has been no improvement over time and then I return the weights to what they were at peak performance, the BSSF.

Graphs

Note: on all graphs, the x-axis is in hundreds of epochs.





Training set accuracy: 93.75

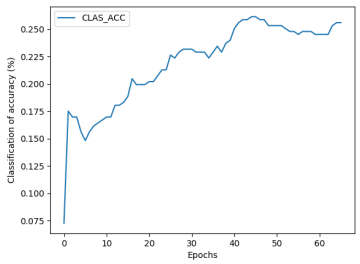
Test set accuracy: 89.47

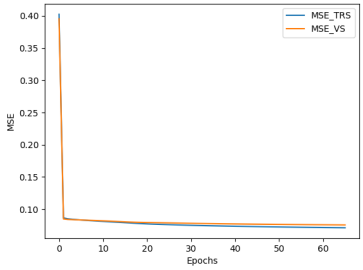
Vowel Classification Problem

Stopping Criteria

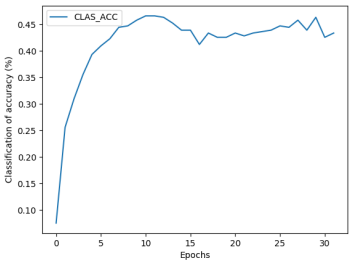
I stopped when the Mean Squared Error (MSE) of the Validation Set (VS) stopped improving over a period of epochs (25). I keep track of the Best Solution So Far (BSSF) and update it after there has been no improvement over time and then I return the weights to what they were at peak performance, the BSSF. (SAME AS ABOVE

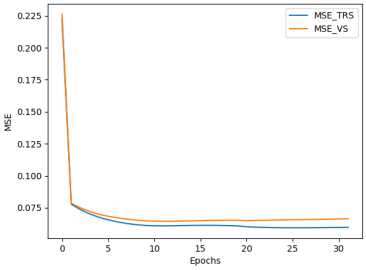
Graphs: learning-rate=0.001, momentum=0.0



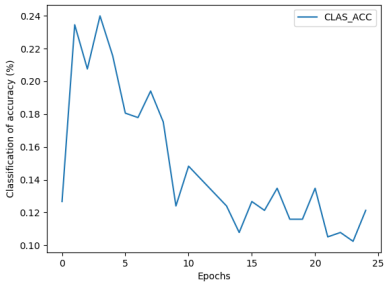


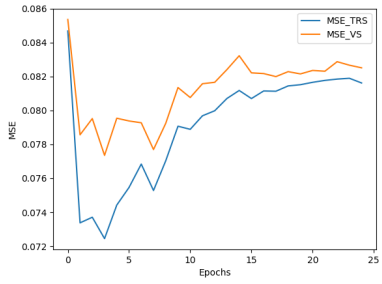
Graphs: learning-rate=0.01, momentum=0.0



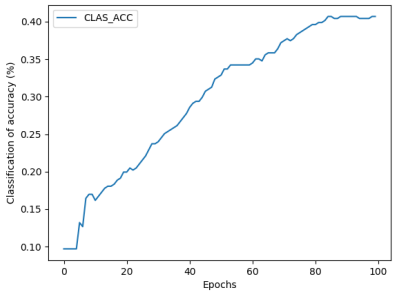


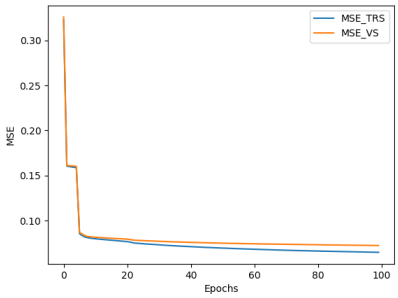
Graphs: learning-rat=0.1, momentum=0.0



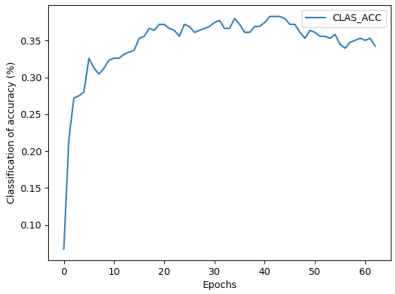


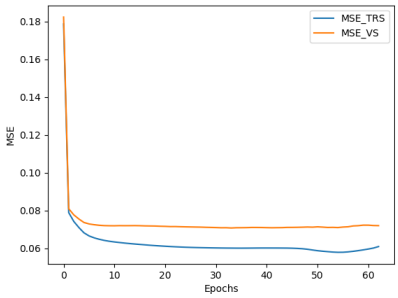
Graphs: learning-rate=0.001, momentum=0.3



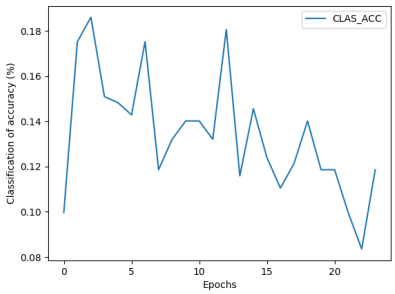


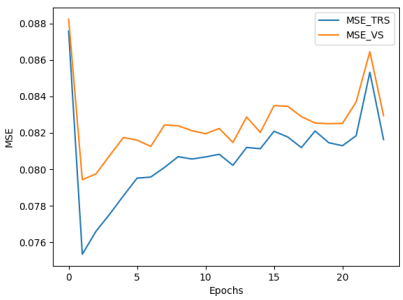
Graphs: learning-rate=0.01, momentum=0.3



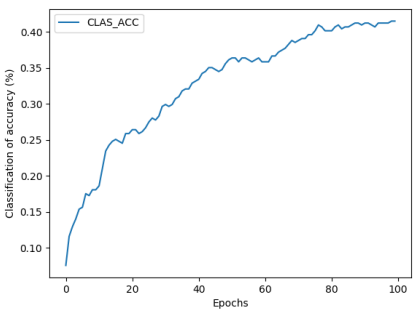


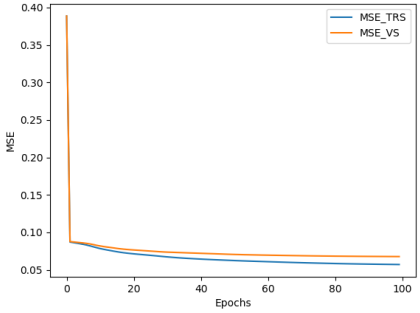
Graphs: learning-rat=0.1, momentum=0.3



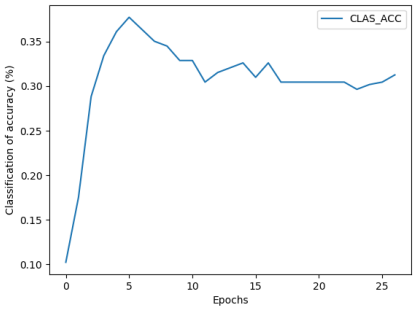


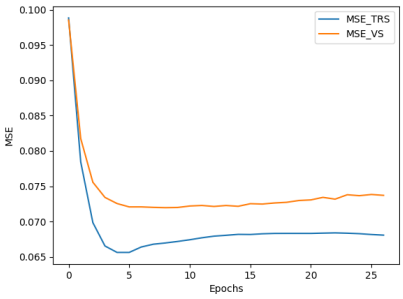
Graphs: learning-rate=0.001, momentum=0.7



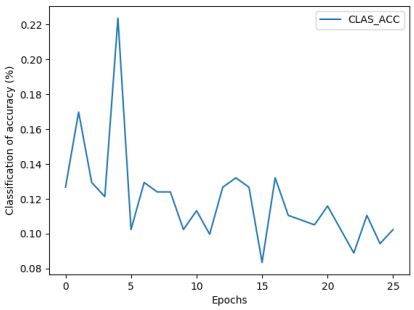


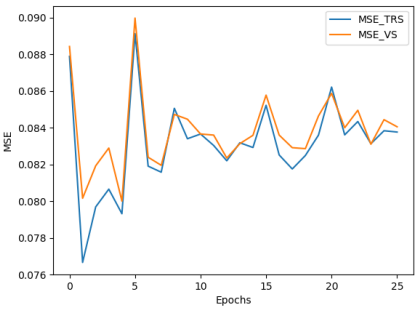
Graphs: learning-rate=0.01, momentum=0.7





Graphs: learning-rat=0.1, momentum=0.7

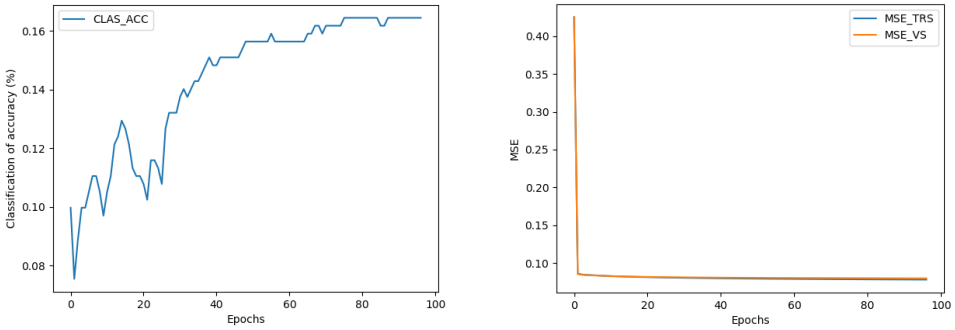




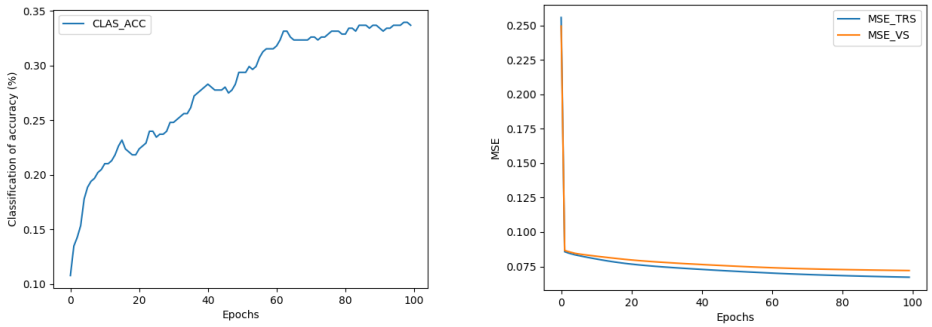
Learning Rate/ Momentum Conclusions

I found that with a learning rate of 0.1 the gradients jump all over the place and it is just too high. A learning rate of .001 is to low and is often trapped in local minima and therefore is also not as good. The final learning rate I tried was .01 which seemed to fair the best. It usually missed local minima but did not fly all over the place. When adding a little momentum, however, a learning rate of .001 outperformed a learning rate of .01 because the momentum was enough to leave the local minima. One interesting observation is that I noticed the amount of momentum made little difference. Small amounts of momentum (.3) allowed the network to speed over local minima as would large mounts (.7). It would seem that you want momentum that is large enough to jump local minima but small enough not to overshoot the better minima. Both .3 and .7 seem to fit this description.

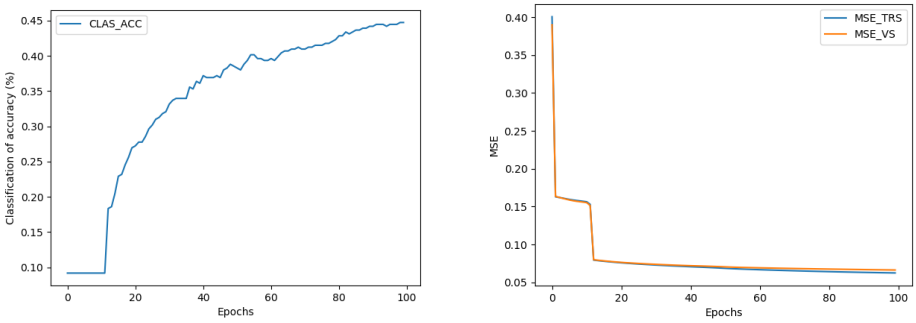
Graphs: learning-rate=0.001, momentum=0.0, hidden nodes: 10



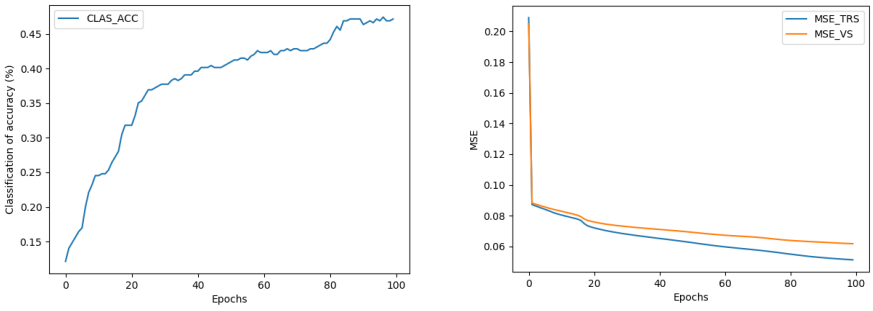
Graphs: learning-rate=0.001, momentum=0.0, hidden nodes: 20



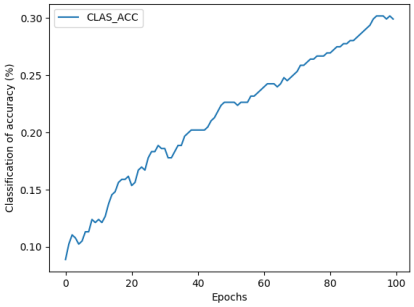
Graphs: learning-rate=0.001, momentum=0.0, hidden nodes: 40

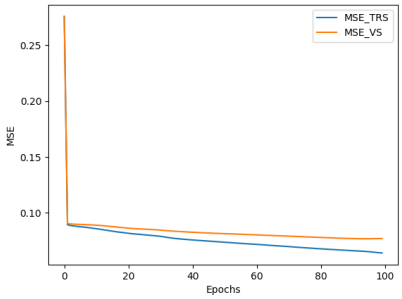


Graphs: learning-rate=0.001, momentum=0.0, hidden nodes: 80

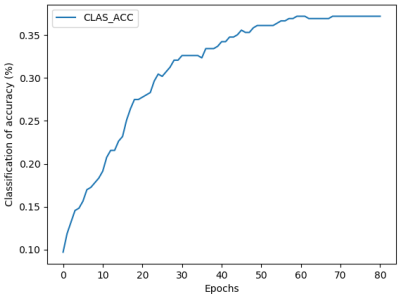


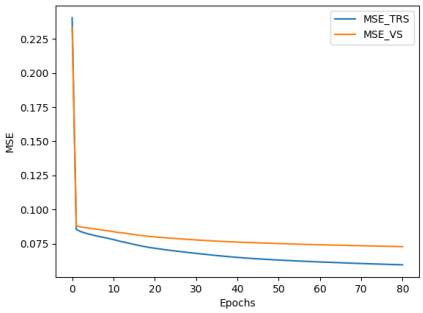
Graphs: learning-rate=0.001, momentum=0.0, hidden nodes: 160





Graphs: learning-rate=0.001, momentum=0.3, hidden nodes: 80





Hidden Node / Momentum Conclusions

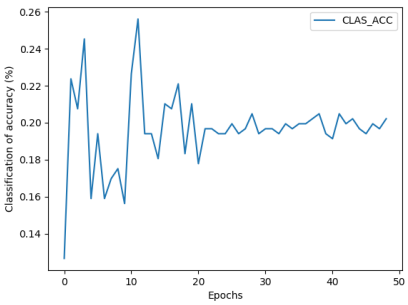
I found that if there are not enough nodes (like in the example of 10) then the capacity to learn in lower, however, if there are too many nodes (160) they can be access and make the network take exponentially longer to train. 80 epochs were the most successful number of epochs in order to produce the best classifier for the vowel dataset. When momentum is added the dataset converges much faster. It also is interesting to note that momentum starts out better than without momentum but it also stops premature. I hypothesis that if you were to change the stopping criteria and allow it to continue to run that momentum would eventually converge to a similar accuracy to the example without momentum.

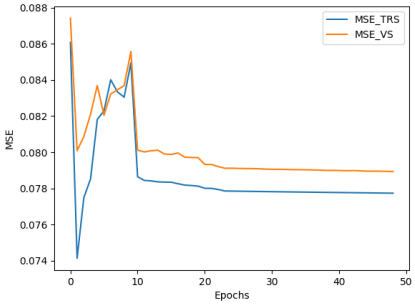
Further Experimentation

I decided to experiment to see what will happen if I try to adjust the learning rate and momentum during execution. My hypothesis is that I will be able to reach a higher accuracy much faster. I think that my current stopping criteria looks for when the network overshoots. Therefore, when we lower the learning rate and the momentum we should see more exact results.

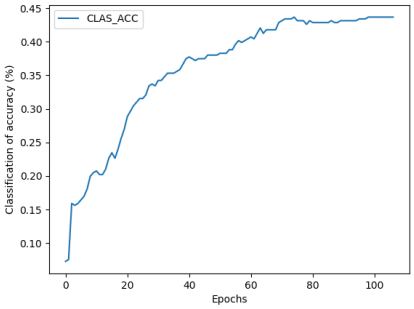
Experiment 1: lowering both the lr and mom

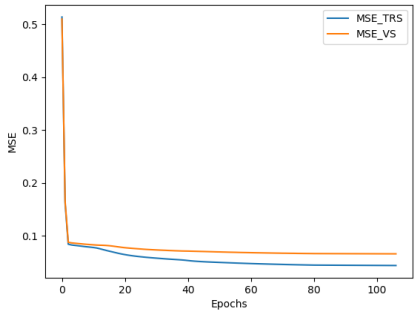
This is my new stopping criteria where k=the number of epochs without change, lr=learning rate, mom= momentum



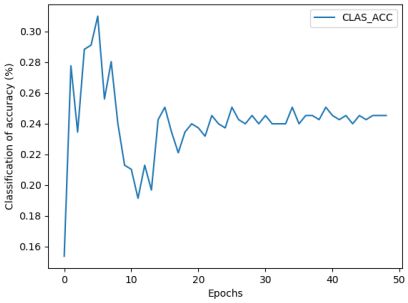


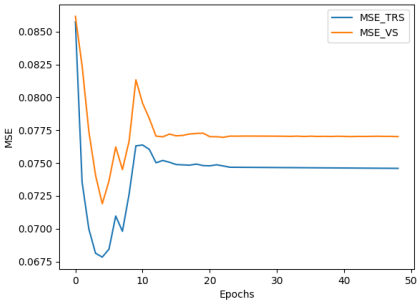
Experiment 2: lowering just the mom





Experiment 3: lowering just the lr



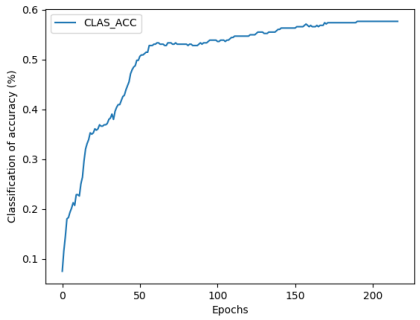


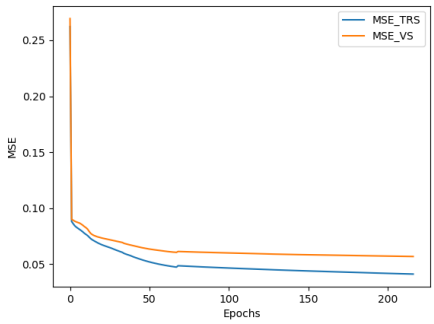
Further Experiment Conclusions

I found that the error caused by overshooting was more than I anticipated. In all 3 experiments the network overshoot and with a decreased learning rate and/or a decreased momentum it was not over to recover from the overshooting. If I was to continue experimenting I would see if jumping to the BSSF and resetting the BSSF would help the network prematurely. I did find that keeping the learning rate constant while lowering the momentum was the most effective.

Experiment 4

I decided to do one more experiment with changing momentum but this time I would jump back to the BSSF after each change in momentum and I would reset the BSSF to avoid stopping prematurely. The results were the best so far!!





Training set accuracy: **65.768%** Validation set accuracy: **57.682%** Test set accuracy: **61.290%**